2004

Data modeling and handling for analysis and visualization in a collaborative setting

Aditya Vaze
University at Buffalo

Vijay Kalivarapu
Iowa State University, vkk2@iastate.edu

Eliot H. Winer
Iowa State University, ewiner@iastate.edu

10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
30 August - 1 September 2004, Albany, New York

AIAA 2004-4603

Data Modeling and Handling for Analysis and Visualization in a Collaborative Setting

Aditya Vaze*
Department of Mechanical Engineering
University at Buffalo, Buffalo, NY, 14260

Vijay Kalivarapu†, Eliot H. Winer‡
Department of Mechanical Engineering
Iowa State University, Ames, IA 50014

Virtual Reality Applications Center
Iowa State University, Ames, IA, 50014

This paper discusses the development of a data modeling and handling methodology to display results from a large-scale Finite Element Analysis in real-time from any geographic location in the world to aid in complex decision-making. The developed methodology enables real-time collaboration before, during, and after a complex engineering analysis. The collaborative capabilities include a three dimensional, interactive representation of the analysis data available through the Internet on any computing platform without the need of installed software or specialized hardware. A scientist has the ability to change data resolutions on-the-fly as well as view animated representations of the analysis results. In this paper, the developed methodology was applied to a geophysical situation. However, the benefits could be realized in a range of application areas from mechanical design to biomedical imaging. The details of the development are presented in this paper. The full paper will present additional descriptions as well as example problems.

I. Introduction

Engineering analyses, though typically applied to mechanical systems or the design of complex products and processes, are increasingly being used in more diverse applications. In areas ranging from geology to medicine, the benefits of applying complex engineering analysis and optimization are being discovered. Processes including medical diagnosis and weather simulation are becoming more efficient, accurate, and robust, in less time for less cost through the application of engineering methodologies. Improving these metrics has been a mainstay of engineering design for many years. The common thread joining all these applications is the data. These data have features such as size, geometry, and discipline specific properties. Whether these properties are flow field velocities or stress calculations, they are simply defining features of the data. These properties are computed from the governing equations used in a particular discipline. Regardless of the exact form of these equations, engineering analysis and optimization can be used to improve these results in a variety of ways.

Once these results are available, a powerful way to effectively assess them is collaboration. Increased computational capabilities and the explosion of the Internet are enabling collaboration to further improve product and process metrics. However, this presents new challenges for engineers. Methods must be developed to efficiently

* Research Assistant, New York State Center for Engineering Design and Industrial Innovation, 5 Norton Hall, Buffalo, NY, 142600, student member.
† Research Assistant, Department of Mechanical Engineering, Virtual Reality Applications Center, 2030 Black Engineering, Ames, IA, 50014, student member.
‡ Assistant Professor, Department of Mechanical Engineering, Virtual Reality Applications Center, 2030 Black Engineering, Ames, IA, 50014, member.

model and handle complex engineering data to enable effective collaboration. This paper presents concepts to handle complex data from a variety of sources and make it usable for real-time web-based, collaboration. This research was applied to a geophysical situation, specifically circumstances around a volcanic eruption. While this is not typically thought of as a engineering problem, it will be shown how core engineering analyses can be used to aid in decision-making for the purposes of risk-mitigation. The goals in this situation are identical to those of more traditional engineering processes, to make decisions that result in accurate results efficiently in as little time as possible. Volcanic eruptions are simulated using Finite Element Analysis (FEA), which has been traditionally used extensively in engineering design processes. In all applications, FEA can produce enormous amounts of data that must be modeled and handled properly for effective use in decision-making.

II. Background

Advances in computer graphics and imaging have changed the way scientists interact with geographic data. Weather updates and predictions have already entered our day-to-day lives and we continue to rely on them, thanks to the capabilities to integrate and analyze geographic data. Before computer based visualization, the prominent mode of communication among the people of this scientific community was traditional 2D mapping. Mapping and geographic data representation techniques are now becoming increasingly digital due to advances in computational power. Today, topographic data can be used in conjunction with various other types of data to study the effect of multiple parameters on a location under investigation. For the first time we have a capability to realistically simulate and estimate the effect of extreme natural circumstances such as earthquakes, volcanoes and tornadoes. Hence, using computer simulations, effective risk mitigation before and during these events becomes a possibility. Typical applications where computer simulations are being used for risk mitigation are flood management, earthquake research, and risk management in the event of a volcanic eruption.

Volcanic eruptions are particularly dangerous phenomena. What makes them so dangerous is a flow containing high-density mixtures of hot, dry rock fragments and hot gases that move away from the vent that erupted them at high speeds. This kind of flow is called a pyroclastic flow. It results from the explosive eruption of molten and/or solid rock fragments. They may also result from the non-explosive eruption of lava when parts of thick lava flow collapse down a steep slope. A second dangerous type of flow is known as a lahar. A lahar is a torrential flow of water-saturated volcanic debris down the slope of a volcano in response to gravity. Pyroclastic flows and Lahars are two of the greatest volcanic hazards. More people have died due to these than any other volcanic hazard[1]. Pyroclastic flows can incinerate, burn, and asphyxiate people and can travel at speeds up to 150 mph[2,3]. Gases within a pyroclastic flow can explode and cause ash to rain down on nearby areas.

One of the most serious issues researchers face in the event of a volcanic eruption is proper risk assessment and management. Many active volcanoes in countries such as Mexico have significant populations located adjacent to or on a volcano. This is due to the rich soil composition found in these areas, which is beneficial for agriculture. If an eruption occurs, significant loss of life, not to mention property, can occur. The speed of a pyroclastic flow, and the close proximity of towns and villages to a volcano, makes immediate response extremely difficult to achieve. Thus, early warning is critical to ensure safety for the populations. A web-based visual tool capable of letting users see how a probable eruption might affect their locality would address many of these issues. Such a tool would merge terrain information, map information, relevant cultural data, and simulated flow data. The flow data would need to be extracted from a large-scale analysis resultant dataset. These datasets are often created using Finite Element Methods (FEM). Advanced FEM codes are used to simulate the movement of pyroclastic flows, landslides (caused by volcanic eruptions), and lahars. These simulations can involve millions of data points per time step and can run for hundreds to thousands of time steps. Currently, the most advanced codes of this nature are highly parallel and run on massive supercomputer architectures.

Because many who will contribute to the risk management decision-making process will be in geographically remote locations, it is critical to develop the capability for them to interact with one another, view simulation results, and discuss possible safety options in a distributed fashion. In addition, there are several levels of communication and dissemination that must be captured (i.e. for the scientists doing the simulations to the public safety officials). The information being used comes from various sources, such as sensors at the remote hazard site (cameras, etc.), GIS data, and immersive simulations. Further, the users themselves will provide yet another kind of information. Using a web-based interface, users can send information to one another, view the information and images, and discuss the data and images – all in real-time. This information is then instantly posted to a common system to be viewed by all others.

III. Problem Definition

The target application for the proposed data model is a web-based, real-time, platform-independent, visualization tool. This tool will be accessible with a common desktop computer, web browser, and a basic Internet connection. This tool will be able to realistically visualize the relevant portions of massive datasets generated by a large-scale, parallel FEM simulation. The fundamental research is in the development of a data model and handling method to accommodate the necessary real-time interaction. The raw datasets, if used in original form, are simply too large to be handled by any visualization or analysis tool over the Internet. The typical file size of these simulations can easily exceed a gigabyte of disk space. For accurate risk mitigation, many simulations must be run and accessible, which combined can easily exceed a Terabyte of disk space. Even using advanced compression techniques, a single simulation would still result in the transfer of up to 500 Megabytes of data. This makes real-time visual interaction simply not possible over most networks and Internet connections. The data for these simulations comes from the following sources:

1. Topography data: This is the information about the terrain and its features (topological, cultural, etc.). All this information is typically provided by certified GIS databases.

2. Flow information: The flow data is obtained from numerical simulation programs. Many researchers around the world are using different programs for these simulations. The current state-of-the-art is massively parallel FEM codes. Examples are the Multi Flow Interface Exchange (MFIX)[4], a Fortran Based program developed by researchers at the National Energy Technology Laboratory (NETL) or TITAN2D[5], a Finite Volume Method based program developed by researchers at University at Buffalo. Flow data is calculated at independent grid locations, determined by an adaptive meshing algorithm. The results from these simulations are most useful if shown aligned properly with topography data. These results provide properties (location, velocity, height) of simulated flows (pyroclastic, etc.) over thousands of time steps. For this work, the TITAN2D program was used to create the simulation datasets.

IV. Research Issues

Based on the needs and requirements outlined, the following research issues were identified:

1. Creation of a portable, flexible data model for the integration of heterogeneous engineering data.
Complex engineering processes, whether the design of a car or simulation of geophysical phenomena, often require multiple data sources to be merged for a single visual representation. This can lead to large, unmanageable datasets. It is necessary to develop a data model that is flexible to accommodate multiple sources of structured and unstructured data. In addition, this model must remain portable to any computing platform or operating system. In addition, special attention has to be provided to maintain accuracy with respect to the originally generated simulation data.

2. Creation of a multi-tiered data structure for real-time, interactive visualization of complex engineering data over the Internet.
High and low fidelity simulation results are necessary for the decision-making in a complex process, such as that for risk mitigation and management under extreme volcanic activity. A single user or multiple users may each require different parts of resultant simulation data to aid their understanding of the situation. Transmitting all the data at once is not possible without significant amounts of time, thus negating real-time interactivity. Thus, a multi-tiered data structure must be developed that allows desired data to be retrieved and viewed in real-time from any location over the Internet. This model has to be developed keeping in mind a target file size, which will depend on the characteristics of a user's connection with the server. In general, the data model should be designed to work with all connections, regardless of bandwidth.

It is important to state that the needs and requirements expressed, although given in terms of geo-spatial datasets and simulations, are extremely similar to those of many other engineering applications. For example, the data generated from a stress analysis of a complex mechanism, also typically done using FEM codes, would have the same requirements for collaborative visualization. The major difference would be the desired end result. In this case, it would most likely be to complete a complex design process in less time for less cost with improved accuracy, efficiency, and/or robustness.

V. Titan2d

To simulate dangerous volcanic flows, high-fidelity analysis has to be performed. This can be done using various numerical approaches such as Finite Difference Methods, Finite Element Methods, or Finite Volume methods. The choice of method depends on various factors such as compressibility of fluid materials, type of flow, and specific assumptions or simplifications, which can be applied. Titan2D is a high-performance computing code for the simulation of pyroclastic flows and landslides over complex terrain. Titan2D is a parallel, adaptive meshing, finite volume computational environment using a Godunov solver for the governing PDEs. One of the highlights of Titan2D is the integration of GIS data to extract topography information dynamically from a Geographic Resource Analysis Support System (GRASS). This provides a realistic simulation method, which exhibits solution adaptability and control of GIS data over numerical and modeling errors. This is critical for accurate, realistic and usable simulations of landslides and volcanic flows. Titan2D also uses a data management scheme that facilitates storage and access to massive and distributed data sets[6].

VI. Data Model Development

The current data model in Titan2D is not sufficient to accurately and efficiently reduce the output dataset of a simulation. A new data model from within the simulation must be developed to handle this. The main emphasis of any work, which aims at the development of platform-independent, web-based visual representations for these types of simulations should address the following requirements:

1. Data Reduction
2. Data Comprehension
3. Interactive Techniques
4. Architecture/Systems

The massive nature of the simulation data is the most limiting issue. This obstructs development of real-time, web-based visualization. So, the primary requirement is to reduce the size of the information that needs to be sent over the Internet with minimal loss of accuracy. The second most important requirement is the identification and location of data, which is essential for complete comprehension of simulation results and its significance. This is connected with data reduction as identification of important and redundant data is inherently going to reduce the size. The third requirement, interactive techniques, allows a user to move seamlessly between different datasets in a simulation to find the information needed. This is very important, as a wide array of datasets, which convey different aspects pertaining to the given terrain, must be available in a simulation. This directly affects the ability of the visualization application to cater to different types of users, each with distinct requirements and interests. Lastly, the Architecture/Systems requirement looks into the core technologies and implementation of the proposed data model. This requirement addresses the arrangement and structure of the simulation data for use by a user.

VII. Data Reduction

Titan2D uses a mesh that adapts to the problem as the solution proceeds. The number of elements at each time step depends on the activity and extent of flow. This is highly unpredictable in nature and so selecting each element for inclusion in a web output file based on its merit and fitness will generate a file size proportional to the number of elements present at that time. Since very stringent constraints on the size of the output file exist, to ensure real-time interaction, this method of data reduction was not pursued. Instead, the number of total grid locations to be included in the output dataset was based on the capacity to transmit data and interact with it in real-time. Assuming the slowest type of Internet connection to be a dial up connection, the size of any data file sent to a client workstation was restricted to be less than 400 KB. This restriction is user-defined and thus, can be altered for any specific situation. This size is independent of the length of the run and the number of elements active in that particular run. Once a simulation run is initiated, a new arrangement of elements and nodes is created and flow is calculated at the active elements for the time step. The arrangement of the elements changes each time-step. In order to extract the flow information, active elements must first be identified. The active elements are the elements that show the presence of flow activity at a particular time step. An element that is active at one time step may become inactive during another time step. The "pile height" parameter indicates the presence of flow at a grid location. If a pile height at any location is non-zero, it indicates the presence of flow mass at that location. All such elements, which have non-zero pile height, are initially stored in memory. The magnitude of the pile height indicates the thickness of the simulated volcanic flow at that location.

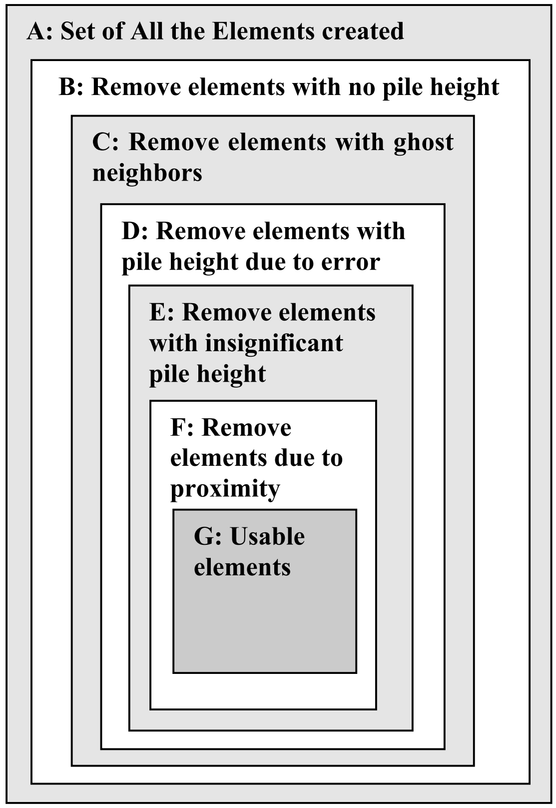


Figure 1 - Elimination of unnecessary data from simulation grid

Once this initial array of node information is compiled, it is processed to identify the locations important for visualization. This was accomplished through a series of checks designed to eliminate points with undesirable properties. The process is shown in Figure 1.

Stage A represents all the elements created at a particular time step. All of them are candidates for flow property calculation. Titan2D then computes flow properties at appropriate points. Once this is complete, the data processing program checks for the existence of pile height at each element. First, all elements without flow activity are eliminated (Stage B). The remaining elements are then candidates for stage C of the elimination process. In this process, all the elements with non-zero pile heights are investigated to check if they have a ghost cell as their neighbor. A ghost cell is an artificial cell element created to maintain simulation accuracy when the mesh is repartitioned, divided, and sent to multiple processors. These cells do not represent actual elements. Thus, if a cell has a non-zero pile height and a ghost neighbor it is discarded, as the pile height does not actually exist. Stage D identifies elements with pile heights due to numerical errors and eliminates them. This computation is determined by the magnitude of the pile height. Using a variety of factors including the length of the run and range of pile height values, an upper limit is calculated. Any pile height value falling below this limit is determined to be from numerical error and is discarded. Stage E is similar to stage D in that elements with pile heights considered computationally insignificant compared to the computed pile heights over the life of the simulation are computed are eliminated. It was observed that pile heights less than 7% of the maximum pile height during the simulation did not contribute to the accuracy of the visual model. Actually, these small pile height values became a distraction to viewers aside from significantly increasing the file size. This brings the data to stage F, which represents real elements with significant flow activity. Further refinement is done at this stage. As the numerical grid is adaptive in nature, during a particular timestep the mesh becomes denser in locations where significant flow activity was present in the previous timestep. This behavior is a major contributing factor in the large amount of data that has to be processed. Having many elements with significant pile height very close to one other is not necessary or efficient for visual representation.

Thus, this stage investigates a particular region where the computational mesh is fine and maintains a few representative points from the many available. The threshold for maintaining or eliminating points is determined from the extent of the computational mesh in 2 dimensions. A ratio is computed between the distance between two adjoining elements and the extent of the domain. A prescribed user parameter then determines if all elements are maintained or some are eliminated. The method described developed several new parameters to aid in data reduction. These parameters were developed to maintain as much visual accuracy with the original data at a fraction of the storage. This methodology could be applied to any analysis involving computations performed on a meshed domain.

VIII. Data Comprehension

Data reduction was just one step towards creating the final data model. Of equal importance was to look at how this information was to be actually represented on a computer. There are various popular methods to show a cloud of discrete points. These methods include creating a volume using those points, showing the outermost points to display the data extents, or creating contours that succinctly show data magnitude information. The main purpose of generating this data model was to enable a remote user to accurately understand overall trends of a simulation It was not aimed at allowing a user to extract precise information about each data element. For example, a geologist should be able to view where a flow is proceeding and how the pile is spreading. Although, a user should be able to focus on a part of a representation and obtain higher fidelity data, there needs to be a tradeoff in the resolution level offered and real-time interaction. The data model proposed will provide the finest resolution of GIS and flow information available while always maintaining real-time interactivity. It was decided that through the use of contouring the visual representation would be enhanced while further reducing the size of the data. After investigating various contouring methods available for a random distribution of points one, which computes small line segments to generate contours was chosen. This method offered the perfect blend of computational efficiency and visual accuracy to be successfully incorporated into the data modeling method. Contour computations were performed on data successively for each time step. First, the data points were sorted into a hierarchical data-structure. This arrangement enabled efficient querying of the data. A sample of the structure is shown in Figure 2.

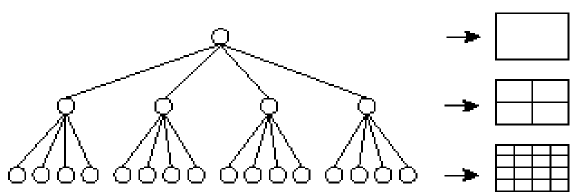
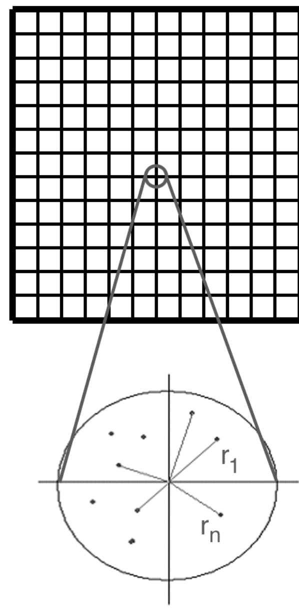


Figure 2 – Hierarchical Data Structure Used for Contouring

The elements from the adaptive grid with significant pile height values were formed into progressively larger groups on varying sub-levels. At each grouping, the maximum and minimum pile height values for all elements in that group are determined. Thus, the tree structure can be searched at each level for a particular pile height value. If the desired value does not fall inside the maximum and minimum for a group, then no other levels below it need to be searched, thus increasing efficiency. Once all the data for a single time step is arranged in this manner, a regular grid is constructed from which the contours will be created.



$$Ph_{node} = \sum_{n} \frac{w_n \bullet Ph_n}{n}$$

Figure 3 - Determining pile height at intermediate grid locations using real elemental information

Figure 3 shows how the construction of the regular grid is performed. A rectangular grid of specified density in two coordinate directions is overlaid onto the adaptive (irregular) computational mesh. Then, each point (node) on the regular grid is examined. A radius is constructed around each regular grid point. All elements from the computational mesh within this radius are used to calculate the pile height at this location. The weighted average of the original pile heights is used to compute the pile height at the regular grid point. The weights are inversely proportional to the distance the original computational element is from the regular grid location. In this manner, pile heights for each point in the regular grid are computed. Then, contours are generated at equal intervals using this regular spaced grid through linear interpolation of the node values to find where a given contour intersects diagonals of the regular grid cells (defined by four nodes). If a contour is determined to pass through a cell, this curve is shown by at least two and a maximum of five line segments, which are drawn inside the cell. Performing the contouring in this manner assures that a given contour is a closed polygon, which represents the flow more accurately. The numerical accuracy of a contour depends on the density of the intermediate grid. This is user specified to accommodate processing speeds or desired accuracy. Once contours are computed, the regular grid is discarded.

IX. Interactive Techniques

One of the most important features of this data model is the ability for a user to change fidelity levels in real-time. Specifically, a user needs to be able to target an area of the simulation and then be instantly presented with higher resolution data. This functionality was accomplished through a hierarchical data structure.

Creation of this hierarchical data arrangement starts with the generation of the lowest level or the coarsest acceptable terrain data. The coarsest resolution level is determined in an automated fashion based on a number of factors including the size of the region selected for simulation (through the Titan2D interface) and the highest fidelity data available. A GIS expert, to ensure accurate terrain characteristics, must first certify the highest fidelity available. The next step is to create the intermediate resolution levels. This is accomplished by using the resolution of the finest data and the coarsest data in conjunction with a requirement on file size. Ultimately, we generate multiple sets of data having different resolution levels. Each set is divided into pre-defined "keypoint" regions, where each keypoint region is supported by a finer resolution data set. Upon viewing the initial data representation (coarsest resolution data), a user is presented with the available keypoint regions, which can be selected. Upon selection, the new region is represented at higher resolution. This higher resolution model of the selected keypoint region has the same number of data points as the previous data level. The new region also has associated keypoint regions that can be chosen for even higher fidelity representations. At each level the user is presented with a smaller geographic area at a higher resolution. Levels are generated until the finest certified data level is reached. Figure 4 shows the arrangement of this hierarchical data structure.

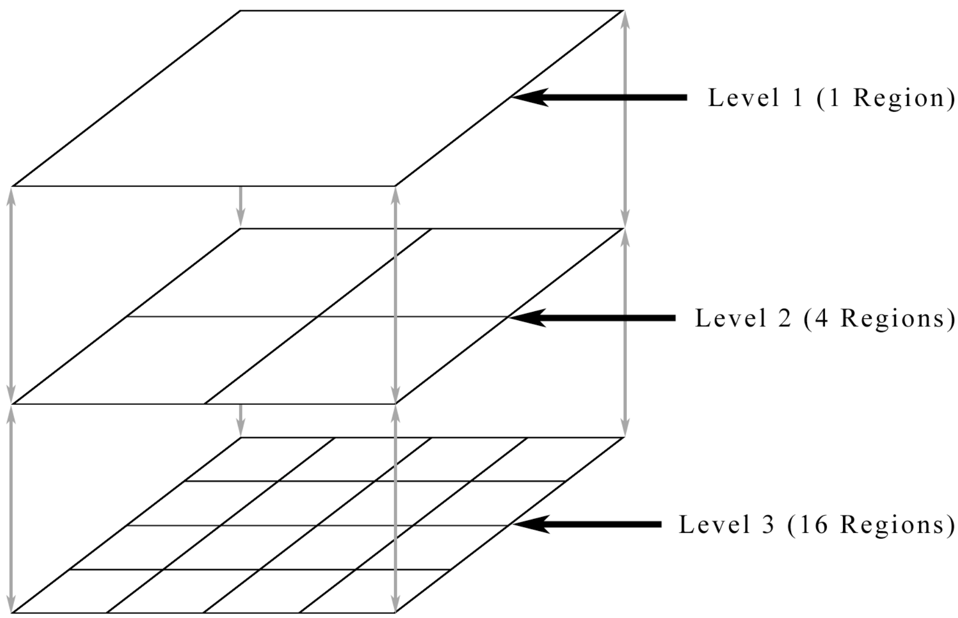


Figure 4 - Structure of data arrangement for keypoint regions

In this arrangement each level (*i*) is divided into a number of regions. For example, in Figure 4 level one (the coarsest resolution) is composed of one region where level three (the finest resolution) is composed of 16 regions. Each region is then decomposed into *k* x *k* number of data points. The total number of keypoint regions m that a level i will have can be computed by:

$$m = n^{n^{i-1}} \qquad (1)$$

In this relationship the coarsest level is 1 and i is incremented by 1 each time the resolution level is increased. At any level, the number of points in any reference direction, N is computed by:

$$N = k \cdot n^{i-1} \qquad (2)$$

To ensure real-time interactivity, the number of data points in a particular keypoint region is held constant. By dividing the area selected for the simulation (from Titan2D) by equation (2), the resolution for a particular level of hierarchy can be determined. Thus, the resolution (R_i) at level i , is calculated by:

$$R_i = \frac{Span}{k \cdot n^{i-1}} \qquad (3)$$

Knowing the finest resolution of data and equations (1-3), the total number of levels that must be created can be computed. Once the total number of levels is computed, independent files are created for each keypoint region.

These files include topography data at the computed resolution as well as flow data at that resolution. These data sources provide the foundation for the transfer mechanism that the system architecture provides.

X. System Architecture

Through a web-based collaborative environment a user is able to access the data sources in real-time. The main issues that were addressed were real-time access to the data and platform-independency. The structure of the keypoint hierarchical data structure ensured that the size of each data source was kept sufficiently small. Based on equations (1-3) a user can pre-set this size or leave the default settings, which set the files to around 300-400Kb. Allowing true platform-independence is another matter altogether. Several technologies for this are currently available including Microsoft Corporation's .NET (http://msdn.microsoft.com/webservices/) environment and Sun Microsystems JavaONE (http://java.sun.com/webservices/index.jsp). Broadly, these technologies encompass what has been termed "web services". Web services are the next generation of Internet based data collaboration. Unlike current web sites, which are mostly static displays of data, web services promises full interactive applications delivered over the Internet. The current implementation of web services involves a steep learning curve to create them (the development environments are very complicated), as well as specific hardware and software requirements on both the client and server computers. Even though web services have been touted as "build software once and deploy it anywhere", in actuality there are a number of issues that must be resolved before this is truly possible.

In this research a modified web services paradigm was used. Essentially, by building software several times (for different platforms) and delivering it through truly platform-independent Internet channels, a user can access a full-featured software application from any computing platform and any geographic location. The delivery method chosen was a web-based system that operates on all workstations and computers and most Personal Digital Assistants (PDAs). The environment is named the Interactive Virtual Environment (IVE). This environment leveraged heavily off work performed by Winer and Bloebaum[7]. IVE runs on most web-browsers including Netscape, Internet Explorer, Mozilla, Safari, Konquerer, and Opera. It was designed to function over a range of Internet connection types from broadband to dial-up always with real-time or near real-time interaction. Figure 5 shows a screenshot of IVE.

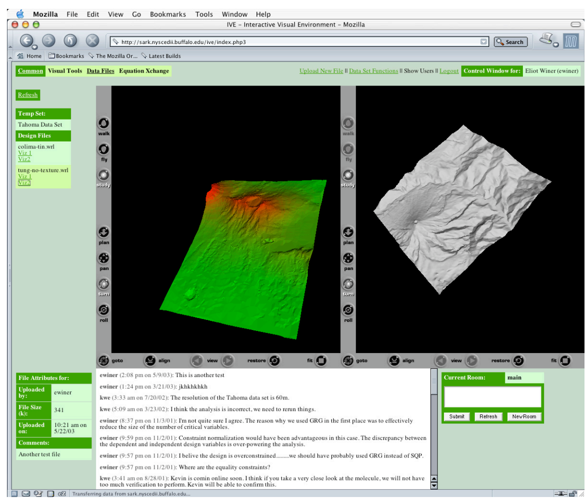


Figure 5 – Collaborative System showing common meeting area

A scientist logging in is presented with various layers in which to work. One layer is the common area (shown in Figure 5). This is where all users can communicate through text-based chat, exchange/view data and share simulation results. A user may also go to a "Visual Tools" layer where the lightweight visual client (QuickView) is available. This client is what displays the data produced by the proposed method in this paper. Here, the user's machine is automatically queried and a QuickView installation is customized for that particular workstation. The software is downloaded to the user's machine, installed (after being granted permission), and launched for use. In this manner, many issues with traditional software installation are eliminated. A third layer provides a workspace to examine files and make notes and observations. This is somewhat offline from discussions and interactions that may be taking place elsewhere. Additional layers for project management and organization are currently being developed. At any time in IVE, a user may view a calendar of events, see who else is logged in and perform detailed data management.

These features allow for data and communication to occur in a variety of channels to boost ideas and improve the quality of decisions made for risk-mitigation. The system has been tested several times with a large number of users (approximately 340) separated by large geographic distances (Columbia, South America to Iowa, United States). In all tests, users all had real-time interaction regardless of Internet connection, computational hardware, or software.

XI. Test Case 1: Mount Tungurahua

Ecuador, a South American country has always been a focus of geologists due to a number of active volcanoes in close proximity to large population centers. One such volcano is Mount Tungurahua (Location: 1.467S, 78.44W) at an elevation of 16,475 ft. (5023 m). The city of Banos, with several thousand residents, lies to the north side of the volcano and is in a high danger zone with respect to volcanic activity. This volcano has approximately 3000 meters of steep vertical slopes. In the event of an eruption pyroclastic flows and lahars can reach the town in minutes. The finest available GIS terrain data is a 30m resolution Digital Elevation Map (DEM). The DEM depicts a region approximately 17km by 12.5 km in area. Thus, elevation coordinates at 236439 evenly spaced locations are known. In ASCII format, the terrain data alone would take up nearly 16 MB of space.

A Titan2D simulation was performed on the Tungurahua GIS dataset. In this simulation the internal friction angle was set to 25 degrees and the bed friction angle was set to 17 degrees. It was a single processor simulation over 1500 time-steps. Results were recorded after every 100^{th} time step. Some of the more pertinent simulation parameters are listed in table 1.

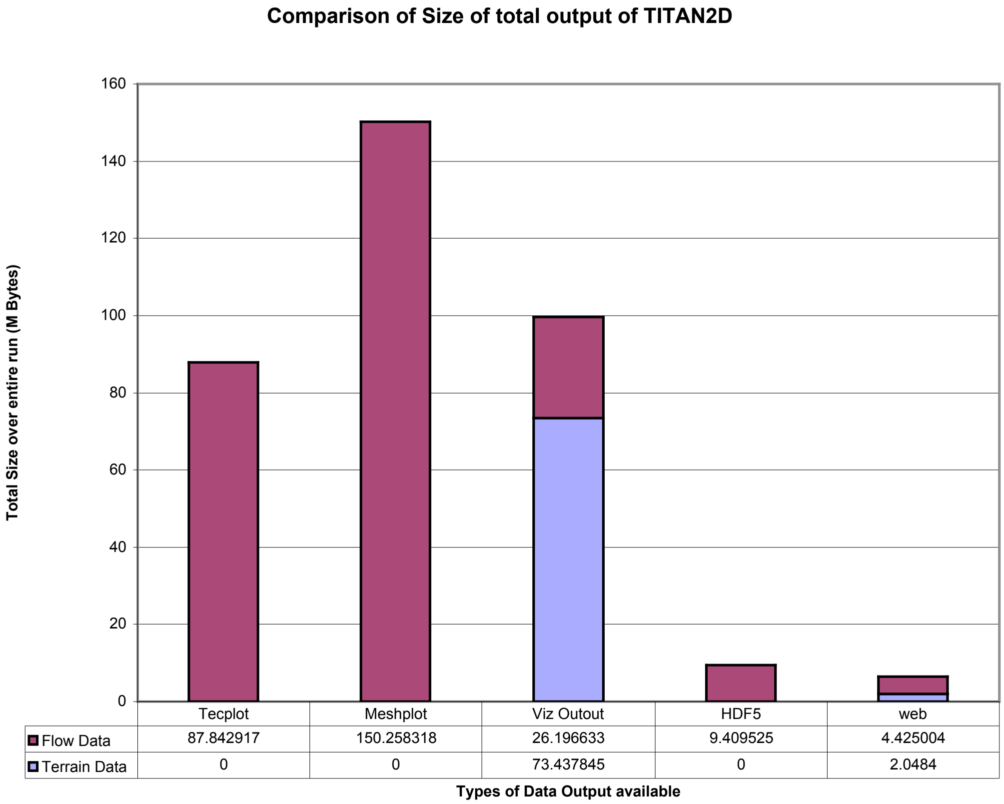
Table 1 - Sample of Input Parameters used for run on Mount Tungurahua.

Number of time step of the entire run	1500
Output taken per time step	100
Total number of outputs	20
Computational Mesh Points in Y direction	200
Initial Thickness of Pile (m)	200
Resolution of the GIS data (m)	30

The output for this run was obtained in all 5 output formats available in Titan2D. These formats are: 1) output for Tecplot visualization software; 2) Mesh Plot output; 3) output for high-end visualization; 4) Hierarchical Data Format (HDF5); and 5) web visualization output. This provided a very good opportunity for comparison. As the output was taken at each 100^{th} time step, there were 15 events when the output was written to data files. HDF5 and high-end visualization output writes all events to a single file. Tecplot, web visualization and mesh plot output writes a single data file for each event, thus each of these output methods creates 15 resulting data files.

Figure 6 shows a comparison of sizes of various data outputs generated during this simulation run. Column 5, titled 'web', refers to the output generated for web-based visualization. It can be clearly observed from Figure 6 that by eliminating the information unsuitable for visualization, and by avoiding the information likely to be repetitive, the size of the total output data has been reduced by a considerable amount. The files that were prepared for the Tecplot software are shown in column 1. These files contain detailed information about all elements in all time steps. However, the impracticality of having such detailed data for web-based visualization can be observed by the magnitude of the total data generated. In addition, the output for Tecplot and Meshplot only shows flow information. These formats do not differentiate between flow and topography data. They provide information about the adaptive mesh at every time step. Thus, these output formats merge the output into a hybrid terrain-flow dataset.

The output data format specifically created for high-end visualization is of particular interest. Column 3 in Figure 6 shows the data size generated for this purpose (labeled "Viz output"). This format writes all the information in a single data file. For purposes of accessing and viewing this data over the Internet, this format would be difficult with which to work. A user would have to download the entire simulation results. Most likely, a user would only be interested in specific regions of the simulation or an overall view. However, the high-end visualization output, as well as the other outputs, only give the user a chance to view different regions after all output data has been accessed. The web output only accesses what a user desires to explore. This was accomplished by the hierarchical structure developed. The data is divided into pieces that are downloaded individually to a client terminal via the Internet. Each piece is self-contained such that a complete visual representation of a region, complete with the flow simulation, can be created. Thus, only data explicitly selected is sent to the client's computational hardware. Overall, the savings achieved using the proposed method was on the order of 90% from the other output forms, except the HDF5 format. However, this format requires software to be installed on the workstation to access the data. This introduces another level of support and issues that must be dealt with. The data created in the web-output format can be easily read and displayed with a lightweight visual client with no other software required.



	Tecplot	Meshplot	Viz Outout	HDF5	web
Flow Data	87.842917	150.258318	26.196633	9.409525	4.425004
Terrain Data	0	0	73.437845	0	2.0484

Figure 6 - Comparison of total size of output generated by each type.

Once the output files were generated, the next phase of post-processing began. In this process an intermediate regular grid was created over the flow data points so that flow contours could be computed. For this stage, only the flow data files were altered. The dimensions of this regular grid were very important because the accuracy of the contours depended directly on it. A fine grid was more accurate but had adverse visual appeal. The lack of visual appeal also hampers quick perception of the data. Through experimentation a compromise between accuracy and visual impact was achieved using a grid dimension range of 40 to 75 points in two coordinate directions (i.e. X and Y). The default value is 50, but a user can modify it as required. A higher number may be used if the flow is spread out over a large area or if it does not form a homogeneous mass. This will ensure that all the important characteristics of the flow will be captured by the contour generated using this grid. Fixing the grid density to a constant value ensures that flow files of relatively the same size are created for every time step. The change in overall size of the simulation data when processed into contours is shown in Figure 7. It is clearly seen that contour post-processing drastically reduces the size of the web output data from TITAN2D for this test case. The overall savings was approximately 40% for various runs performed on this data set. This is in addition to the savings achieved just by using the web output format. So, for this test case where the outputs ranged from 100-150MB, the web output form was reduced to approximately 2-3MB for visual representation. In addition, this 2-3MB is not all sent to the client in one piece, but rather it is sent in smaller pieces as requested as described earlier.

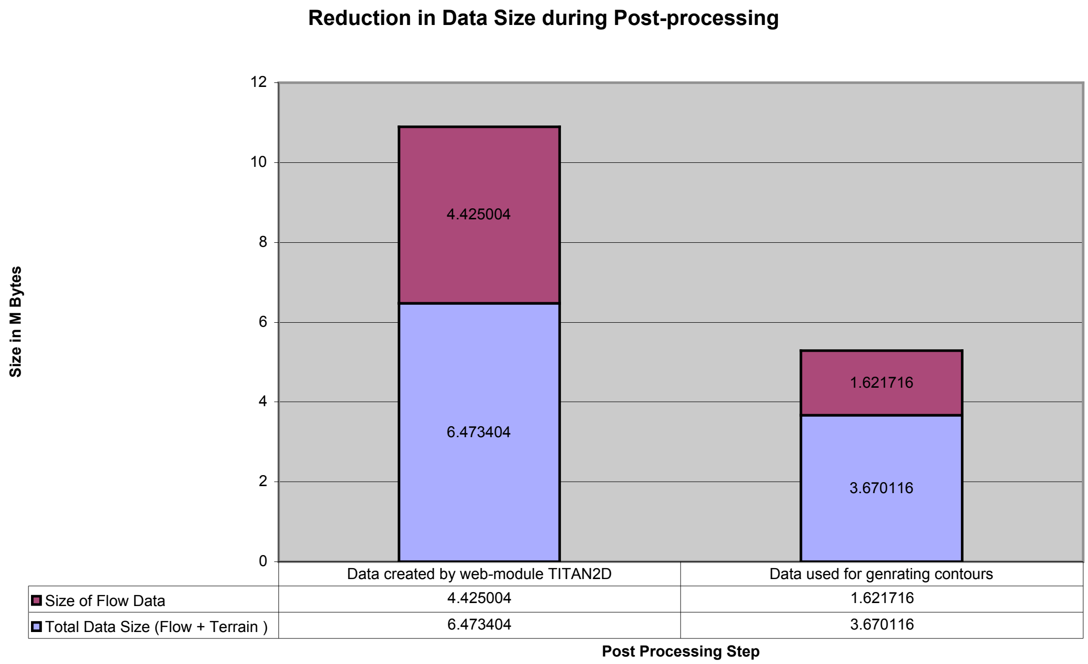


Figure 7 - Data Size Reduction During Post Processing

The original computational grid used in TITAN2D was dense near or under the flow due to its adaptive nature. Thus, all the data points representing the flow mass were very closely packed together. When these points were processed through the developed methodology and into the proposed data structure, many unnecessary and redundant points were eliminated. These results are shown in Figures 8-11. These figures show screen captures of the data represented in the QuickView visual client accessible through the IVE. Data created from the developed method was used in this environment to allow researchers and scientists to access these simulation results from any geographic location from any type of computational hardware. The goals of platform-independent, real-time access were achieved for this test case. The simulation data was delivered on workstations running Linux, MacOS, Windows 2000, Windows XP, Solaris, and Irix. These workstations had differing types of Internet connections including dial-up at 56K, wireless 10Mps, Cable Modem, T1, and T3 connections. The complete test specifications and access times are provided in Table 2. In all cases, the data was delivered to the client in less than 1 minute. The only exception was some dial-up connections, which took between 2-3 minutes to complete.

Table 2 - End user hardware test results

Hardware Specifications	Network Connection	Access Time (seconds)
Windows 2000	Cable Modem	32
Windows XP	10Mbs	21
MacOS X	802.11g Wireless	15
Solaris	T3	12
Irix	T3	14
Linux	T1	19
MacOS X	802.11b Wireless	45
Windows XP	56K Dialup	147

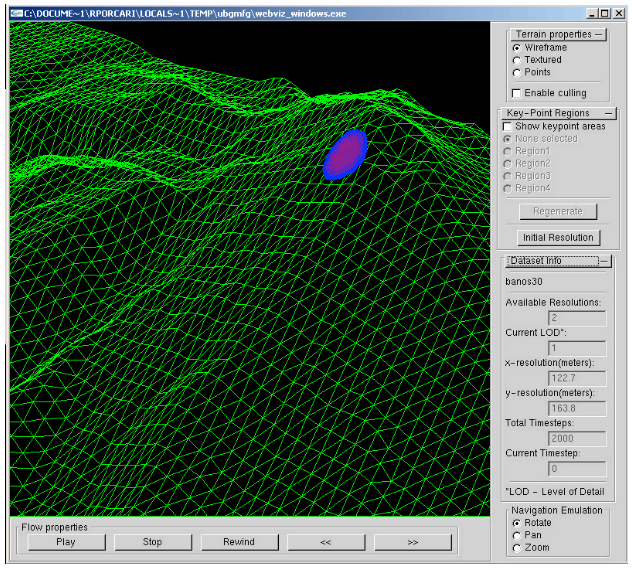


Figure 8 – Visual client showing tessellated Tungurahua terrain with mass flow

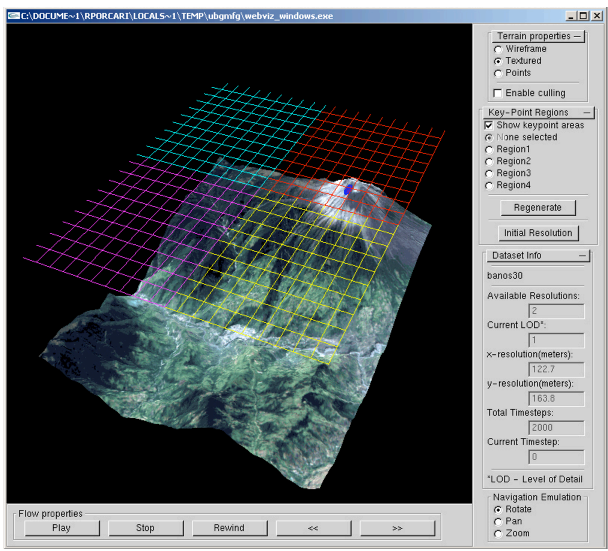


Figure 9 – Visual client showing textured Tungurahua terrain with mass flow and keypoint regions displayed for selection

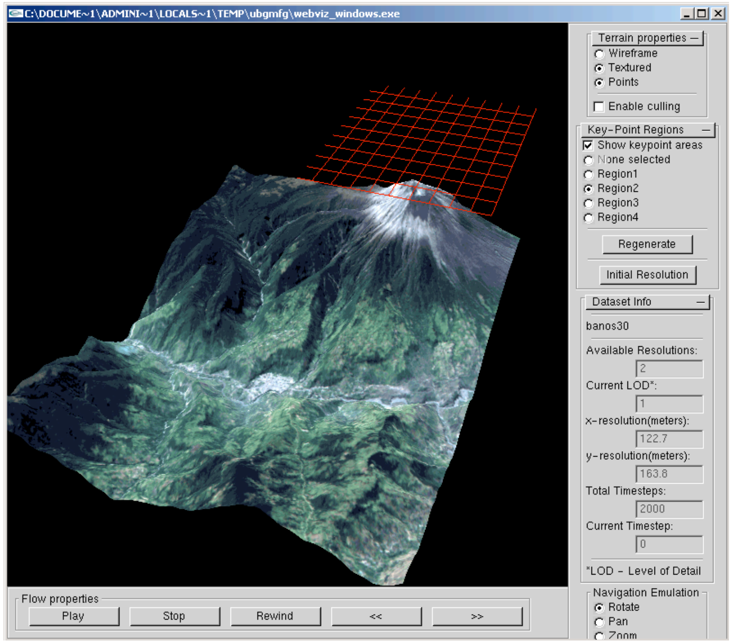


Figure 10 - Tungurahua with Key-Point region containing Flow highlighted

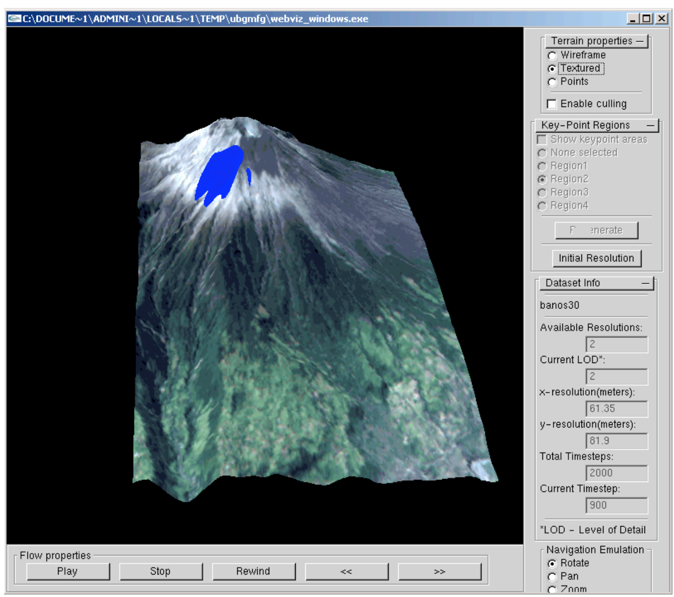


Figure 11 - Tungurahua with Key-Point region of flow in display at twice initial resolution

XII. Conclusions

In this paper a method for structuring and handling large amounts of data from an FEM analysis was presented. The proposed method is a multistage algorithm that involves innovative methods of data reduction, data comprehension, and interactive techniques. This combined with a unique system architecture provides real-time, platform-independent access to complex data in a detailed, three-dimensional representation from any geographic location. No special software or hardware requirements are necessary to access or interact with this data representation.

While this work has shown promising results there are still several key areas of improvement. The incorporation of formal decimation and compression techniques may further reduce the size of resultant data. This would improve real-time access or allow more data to be retained, if necessary. Also, many of the steps need be generalized to provide a formal theory for using this data structuring approach on any FEA.

References

[1] Chester, D., *Volcanoes and Society*, Routledge, Chapman and Hall, Inc., New York, 1993

[2] Tilling, Topinka, and Swanson, "Eruptions of Mount St. Helens: Past, Present, and Future", USGS General Interest Publication, 1990

[3] Hoblitt, Miller, and Scott, "Volcanic Hazards with Regard to Siting Nuclear-Power Plants in the Pacific Northwest", USGS Open-File Report 87-297, 1987

[4] D'Azevedo, E., Syamlal, M., Pannala, S., and O'Brien, T., "Hybrid (OpenMP and MPI) parallelization of MFIX: a multiphase CFD code for modeling fluidized beds", Proceedings of the 2003 ACM symposium on Applied computing, ACM Press, New York, NY, USA, 2003

[5] Patra, Bauer, Nichita, Pitman, Sheridan, Bursik, Rupp, Webber, Stinton, Namikawa, Renschler." Parallel Adaptive Numerical Simulation of Dry Avalanches over Natural Terrain." In review for the Journal of Volcanology and Geothermal Research, 2004

[6] Sheridan, M., Bloebaum, C.L., Kesavadas, T., Patra, A., and Winer, E.H.," Visualization and Communication in Risk Management of Landslides, Proceedings of Risk Analysis, Sintra, Portugal, June 19-21, 2002

[7] Winer, E.H., Bloebaum, C.L., "Using the World Wide Web to Employ Concurrent Design Methodologies", Proceedings of 3rd World Congress of Structural and Multidisciplinary Optimization (WCSMO-3), Amherst, NY, May 17-21, 1999